Características de los sistemas concurrentes

1. Orden de ejecución de las instrucciones
   - SECUENCIAL---ORDEN TOTAL
     - ANTE UN CONJUNTO DE DATOS DE ENTRADA SE SABE SIEMPRE PARA
     - DONDE VA EL PROGRAMA
   - CONCURRENTE--- ORDEN PARCIAL
     - NO SE PUEDE SABER CUAL ES EL FLUJO DE EJECUCION
     - COBEGIN
       - SI, S2, S3, S4
     - COEND
2. Indeterminismo
   - El orden parcial lleva a que los programas tengan resultados diferentes, (indeterminista)

Ejemplo de un programa no determinista

```
Program Incognita
        var x: integer

process P1:
    var i : Integer
    begin
        for i: 1 to do 5 x:=x+1;
    end

process P2:
    var i:Integer
    begin
        for i:1 to do 5 x:=x+1;
    end
```

```
Begin
   x:=0
    cobegin
        P1;
        P2;
    coend
End
```

x=x+1

P1	P2
LOAD x R1 - LPI	LOAD x R1 - LP2
ADD R1, 1 - AP1	ADD R1, 1 - AP2
STORE R1, x - SP1	STORE R1, x - SP2

Posibles secuencias de ejecución

LP1,AP1,SP1 - LP2,AP2,SP2 – …. OK

LP1,LP2 - AP1,AP2 - SP1,SP2 ….. OK

LP1,AP1 - LP2,AP2 – SP1, SP2 ……. NOT OK

Evaluando por condiciones de Bernstein

- S1 -> x=x+1;
- S2 -> x=x+1;

- L(s1) = x E(s1) = x
- L(s2) = x E(s2) = x

- $L(s1) \cap E(s2) = x \neq \emptyset$
- $E(s1) \cap L(s2) = x \neq \emptyset$
- $E(s1) \cap E(s2) = x \neq \emptyset$

NO CUMPLE LAS 3

- De lo anterior, la ejecución de las instrucciones (LOAD, ADD, STORE) debe ser indivisible/atomica
- S1 y S2 no pueden ejecutarse concurrente (por Bernstein) los conjuntos de escritura y lectura no son disjuntos
- Lo anterior implica que S1 y S2 estan accediendo a una variable compartida
- Cuando una variable compartida se lee y se escribe por más de un proceso se tiene una region CRITICA.